Tópicos em Engenharia de Software: Relatório Técnico

Izabella de Castro Lucas

https://github.com/Izabella-Castro/Pratica-4

Introdução

A atividade realizada demonstra como realizar testes de código em Python no contexto de processamento de dados, com ênfase nas boas práticas de engenharia de software. Testar o software é uma parte fundamental do desenvolvimento de qualquer aplicativo, mas a importância dos testes se torna ainda mais crítica em ambientes de Big Data.

O processamento de dados em Big Data envolve a manipulação e análise de conjuntos de dados extremamente grandes e complexos. Pequenos erros ou falhas no código podem ter um impacto significativo nos resultados, levando a análises imprecisas ou mesmo a interrupções nos processos de negócios.

1 Desenvolvimento de Testes Unitários

1.1 Função avgAgeCountry

Descreva o processo de implementação dos testes para a função ‘avgAgeCountry‘, incluindo os cenários considerados:

- Arquivo JSON vazio.
  Neste teste, foi criado um arquivo JSON vazio (”empty_data.json”) e carrega o arquivo usando a função read_json_file. Em seguida, é chamado a função avgAgeCountry com o país ”US” e verifica se a média de idade é igual a 0, uma vez que não há dados para o país ”US” no arquivo vazio.

- Valores de idade ausentes ou nulos.
  No teste test_avgAgeCountry_with_missing_age, é criado uma lista de dicionários de pessoas em que alguns deles têm valores de idade ausentes (definidos como None). Em seguida, é calculado a idade média para o país ”US” usando a função avgAgeCountry. O teste verifica se a idade média é igual a 0, uma vez que os valores de idade estão ausentes ou nulos.

- Campo ‘country‘ ausente ou nulo.
  No teste test_avgAgeCountry_with_missing_country, é criado uma lista de dicionários de pessoas em que alguns deles têm o campo ”country” ausente ou definido como None. Em seguida, é calculado a idade média para o país ”US” usando a função avgAgeCountry. O teste verifica se a idade média é igual a 0, uma vez que o campo ”country” está ausente ou nulo.

1.2 Outras Funções de Processamento

Foi realizado a implementação da função avgAgeCountry no arquivo dataProcessor.py. Nesta função é realizado o cálculo da média de idades para o país desejado. No código, a função avgAgeCountry aceita três argumentos:

- data: A lista de pessoas representadas com campos como "name", "age", e "country"
- country: O país para o qual deseja calcular a idade média.
- age_transform_func: Uma função de transformação opcional que pode ser usada para pré-processar a idade antes de calcular a média. Esta função é útil se deseja calcular a média de idades em unidades diferentes (por exemplo, meses em vez de anos).

1.3 Função de Transformação

Como a função criada já possui uma estrutura para receber qualquer tipo de valor, não foi necessário realizar modificações em sua estrutura.

No teste test_avgAgeCountry_with_age_in_months, é usando uma função age_in_months para transformar a idade de anos para meses antes de calcular a média. O teste verifica se a média de idades em meses é calculada corretamente para o país desejado.O resultado é então verificado usando self.assertEqual.

2 Objetivos dos Testes

Descreva os objetivos por trás dos testes desenvolvidos. Aborde:

- Tipos de problemas que estavam sendo prevenidos:
  Os testes desenvolvidos visam garantir a robustez e a flexibilidade da função avgAgeCountry e prevenir uma série de problemas, incluindo a integridade dos dados, a capacidade de filtragem e a capacidade de aplicar mudanças.
- Relevância do teste unitário e a liberdade para criar diferentes asserts:
  A relevância dos testes unitários está em garantir que a função avgAgeCountry atenda aos requisitos específicos e lide corretamente com diferentes situações e tipos de dados. Os testes unitários permitem uma abordagem granular para verificar se cada parte do código funciona como esperado, o que é fundamental para manter a qualidade do software.

  Além disso, a liberdade para criar diferentes assertivas é importante porque permite que os testes sejam adaptados a diferentes cenários e requisitos. Os testes podem ser personalizados para verificar não apenas a correção da função, mas também se ela atende aos critérios de negócios específicos. Isso é particularmente importante em ambientes de Big Data, nos quais a manipulação e análise de dados podem variar amplamente.

– A intenção de cada teste:
test_avgAgeCountry_no_data_for_country: O objetivo deste teste é verificar como a função se comporta quando não há dados para o país especificado. Isso previne problemas relacionados à ausência de dados relevantes para o país, como valores nulos ou campos ausentes.

test_avgAgeCountry_with_missing_age: Este teste visa garantir que a função consiga lidar com registros que possuem valores de idade ausentes (None) e que calcule a média de idade corretamente, ignorando esses registros. Isso é relevante para prevenir problemas relacionados à integridade dos dados.

test_avgAgeCountry_with_missing_country: A intenção deste teste é assegurar que a função lide adequadamente com registros que possuem o campo "country" ausente (None) e que calcule a média apenas para os registros com o país especificado, ignorando os registros com país ausente. Isso previne problemas relacionados à filtragem de dados por país.

test_avgAgeCountry_with_age_in_months: Este teste visa verificar se a função consegue aplicar uma função de transformação à idade antes de calcular a média. Neste caso, a função age_in_months converte as idades de anos para meses. A intenção é garantir que a função seja flexível e permita a aplicação de transformações personalizadas na idade antes de calcular a média.

3 Reflexão sobre Testes em Big Data

Escrever testes para ambientes de Big Data apresenta desafios específicos devido à escala massiva e complexidade dos dados envolvidos. Uma das principais dificuldades está na geração de dados de teste que se aproximem das características reais, o que requer conjuntos de dados significativamente grandes e diversificados.Além disso, garantir a consistência e a qualidade dos dados é fundamental, pois erros em larga escala podem ser custosos.

Testar funções que processam grandes volumes de dados é de suma importância, especialmente quando se lida com otimização de desempenho usando PySpark. Essas funções desempenham um papel crucial na análise e no processamento eficiente de dados em larga escala, com implicações diretas na tomada de decisões e na qualidade das saídas. Testes rigorosos garantem que as funções processem os dados corretamente, mantendo a integridade e a consistência dos resultados, mesmo em cenários de alta complexidade. Além disso, ao considerar a otimização de desempenho com PySpark, os testes ajudam a identificar gargalos, ineficiências e possíveis melhorias no código, permitindo que o processamento de grandes volumes de dados seja feito de forma mais rápida e econômica. Assim, a combinação de testes de qualidade e otimização de performance com PySpark assegura que as funções de processamento de dados sejam confiáveis, escaláveis e eficientes, tornando-se fundamentais em ambientes de Big Data, nos quais a precisão e a velocidade são críticas.

Para realizar um teste de desempenho no cenário da atividade da média das idades, primeiro, é importante definir um conjunto de dados (users.json) de tamanho sig-

nificativo que simule condições de uso realistas. Esse conjunto deve incluir uma variedade de idades e países para abranger diferentes cenários. Em seguida, utilizar uma biblioteca de teste de desempenho, como pytest-benchmark no Python, para medir o tempo que a função avgAgeCountry leva para calcular a média em relação ao tamanho do conjunto de dados. Os resultados podem ser usados para identificar possíveis gargalos de desempenho, permitindo otimizações e ajustes no código, se necessário. Além disso, o teste de desempenho pode ser executado em vários ambientes de hardware e configurações do PySpark para avaliar como a função se comporta sob diferentes condições.

4 Conclusão

Durante a atividade, ficou evidente a importância crítica dos testes no desenvolvimento de funções de processamento de dados, especialmente em ambientes de Big Data. Os testes desempenham um papel fundamental na garantia da qualidade, robustez e confiabilidade das funções, uma vez que permitem a validação da corretura do código em diferentes cenários. Através dos testes unitários, pudemos abordar diversos casos, desde dados ausentes até transformações personalizadas, assegurando que a função avgAgeCountry seja capaz de lidar com uma ampla gama de situações. Além disso, exploramos a importância de testes de desempenho para avaliar o desempenho da função em grandes volumes de dados, identificando potenciais áreas de otimização.

As principais conclusões desta atividade ressaltam que testar funções de processamento de dados é um componente crítico do desenvolvimento de software, especialmente em ambientes de Big Data, nos quais a integridade, a escalabilidade e o desempenho são essenciais. Além disso, a flexibilidade para criar diferentes asserts em testes unitários é fundamental, uma vez que permite a adaptação dos testes a requisitos específicos. Ao abordar diversos cenários, desde dados ausentes até transformações personalizadas, garantimos que a função seja resiliente e atenda a diversas necessidades. Em suma, a atividade destaca que os testes desempenham um papel central na garantia da qualidade do código de processamento de dados, tornando-o mais robusto e confiável em ambientes de Big Data.